IN THE CLAIMS:

Please amend the claims as follows:

1. (Currently Amended) A pipeline, comprising:

a plurality of operational stages, the stages including:

a pointer register stage which stores pointer information and updates;

a pointer dependency checking stage located ~~downstream~~ upstream of the pointer register stage, which determines if instruction pointer dependencies exist and stalls an instruction ~~prior to issuance~~ if necessary to resolve inter-instruction dependencies in the pointer register stage;

a dependence checking stage located downstream of the pointer register file and configured to perform checking on instructions and stalling an instruction if necessary to resolve inter-instruction dependencies in an instruction register file;

an issue stage located downstream from the dependence checking stage;

the instruction register file configured to store instruction information and updates;

at least one pointer functional unit providing pointer information updates directly to the pointer register stage and/or directly to an input of the at least one pointer functional unit as a pointer bypass such that the pointer information is processed and updated; and ~~to the pointer register stage before an instruction goes through the dependency checking stage, an issue stage, and an execution stage~~

at least one instruction functional unit providing information updates directly to the instruction register stage and/or directly to an input of the at least one instruction functional unit as an instruction bypass such that instruction update information is processed and updated and wherein instructions updating the instruction register file and instructions updating the pointer register file are simultaneously processed to track and keep current pointer and dependency updates between instructions.

2. (Currently Amended) The pipeline as recited in claim 1, ~~further comprising~~ wherein the at least one pointer functional unit is located between the pointer register stage and ~~a pointer execution stage used before~~ the dependence~~y~~ checking stage such that inter-instruction dependency is checked after the at least one pointer functional unit~~execution stage~~ or in parallel with pointer execution.

3. (Original) The pipeline as recited in claim 2, further comprising a path for making pointer updates available to the pointer register stage before the instruction reaches a write back stage of the pipeline.

4. (Currently Amended) The pipeline as recited in claim 3, wherein the path includes a normal pointer update path which returns pointer information from the at least one pointer functional unit.

5. (Currently Amended) The pipeline as recited in claim 1, ~~3, further comprising~~ wherein the pointer ~~execution~~ bypass ~~making~~ makes pointer updates available to immediately following instructions before a pointer update is written into the pointer register file.

6. (Original) The pipeline as recited in claim 2, further comprising a pointer reorder

buffer coupled to the pointer register stage to maintain a precise state of pointers.

7. (Original) The pipeline as recited in claim 6, further comprising a precise pointer file for storing the precise state of the pointer reorder buffer.

8. (Original) The pipeline as recited in claim 7, further comprising an interrupt recovery path which keeps the pointer register stage up to date with reordering or recovery information from the precise pointer file.

9. (Original) The pipeline as recited in claim 1, further comprising a combined pointer reorder buffer/issue stage coupled to the dependence checking stage to issue instructions and maintain a precise state/order of pointers.

10. (Original) The pipeline as recited in claim 9, wherein the combined pointer reorder buffer/issue stage includes a table with a plurality of fields to keep identifiers of all pointers that are updated by an instruction, and new values of the updated pointers.

11. (Original) The pipeline as recited in claim 9, further comprising a precise pointer file for storing the precise state of the combined pointer reorder buffer/issue stage.

12. (Original) The pipeline as recited in claim 11, further comprising an interrupt recovery path, which restores the pointer register stage to the precise state from the precise pointer file.

13-21. (Canceled)

22. (Currently Amended) A method for updating pointers ahead of an instruction, comprising the steps of:

providing a plurality of operational stages, including a pointer register stage which stores pointer information and updates, a pointer dependence checking stage located upstream~~downstream~~ of the pointer register stage, which determines if instruction dependencies exist and stalls an instruction ~~issue prior to issuance~~ if necessary to resolve inter-instruction dependencies in the pointer register stage, a dependence checking stage located downstream of the pointer register file and configured to perform checking on instructions and stalling an instruction if necessary to resolve inter-instruction dependencies in an instruction register file configured to store instruction information and updates, an issue stage located downstream from the dependence checking stage, and at least one pointer functional unit providing pointer information updates directly to the pointer register stage and/or directly to an input of the at least one pointer functional unit as a pointer bypass, and at least one instruction functional unit providing information updates directly to the instruction register stage and/or directly to an input of the at least one instruction functional unit as an instruction bypass; and

processing instruction dependency information and pointer information to update ~~the pointer information for~~ the instruction register stage and the pointer register stage so that ~~updated pointer information is available~~ instructions updating the instruction register file and instructions updating the pointer register file are simultaneously processed to track and keep current pointer and dependency updates between instructions.

~~such that the pointer information is processed and updated to the pointer register stage before an instruction goes through the dependency checking stage, an issue stage, and an execution stage.~~

23. (Currently Amended) The method as recited in claim 22, further comprising a step of providing pointer updates to the pointer register stage via an early pointer update path by providing the at least one pointer functional stage~~a pointer execution stage used before~~ between the pointer register stage and the dependence checking stage.

24. (Original) The method as recited in claim 23, further comprising a step of maintaining a precise state of pointers by employing a pointer reorder buffer.

25. (Original) The method as recited in claim 24, further comprising a step of storing the precise state of the pointer reorder buffer in a precise pointer file.

26. (Original) The method as recited in claim 24, further comprising updating reordering or recovery information from the precise pointer file using an interrupt recovery path to the pointer register stage.

27. (Currently Amended) The method as recited in claim 23, further comprising maintaining a precise state/order of pointers using a combined pointer reorder buffer/issue stage coupled to the ~~rename and~~ dependence checking stage.

28. (Currently Amended) The method as recited in claim 27, wherein the combined pointer reorder buffer/issue stage includes a table with a plurality of fields ~~to~~, and keeping identifiers of all pointers that are updated by an instruction, and new values of the updated pointers in the fields.

29. (Original) The method as recited in claim 27, further comprising storing the precise state of the combined pointer reorder buffer/issue stage using a precise pointer file.

30. (Currently Amended) The method as recited in claim 29, further comprising providing an interrupt recovery path which keeps the pointer register stage up to date with reordering or recovery information from the precise pointer file.